

# 变量运算符

# 目录

---

- ◆ 标识符
- ◆ 关键字
- ◆ 变量
- ◆ 基础数据类型
- ◆ 运算符

## ▶ 标识符命名规则

凡是可自己取名的，都叫标识符！都必须守规则！

### ▶ 定义：

Java对各种变量,方法和类等要素命名时使用的字符序列称为标识符

### ▶ 规则：

标识符应以字母,下划线,美元符号开头

标识符应以字母,下划线,美元符号或数字组成

标识符对大小写敏感,长度无限止

### ▶ 合法的标识符：

Identifier、username、user\_name、\_userName、\$username

### ▶ 非法的标识符：class、98.3、Hello World

- ▶ 一些被赋以特定含义，用做专门用途的字符串称为关键字(keyword)

abstract	do	implement	private	this
boolean	double	import	protected	throw
break	else	instanceof	public	throws
byte	extend	int	return	transient
case	false	interface	short	true
catch	final	long	static	try
char	finally	native	strictfp	void
class	float	new	super	volatile
continue	for	null	switch	while
default	if	package		synchronized

- ▶ 一般关键字都为小写
- ▶ 一般编译器都会将关键字以特殊颜色标出
- ▶ Java无sizeof, goto, const关键字，但已做为保留关键字

## ▶ 定义:

- ▶ 变量就是一个由标识符命名的数据项
- ▶ 变量就是系统为程序分配的一块内存单元，用来存储各种类型的数据。
- ▶ 变量一定要先声明，再使用

## ▶ 变量名必须是一个合法的标识符

## ▶ 变量的声明：类型 名称[ = 初始值];

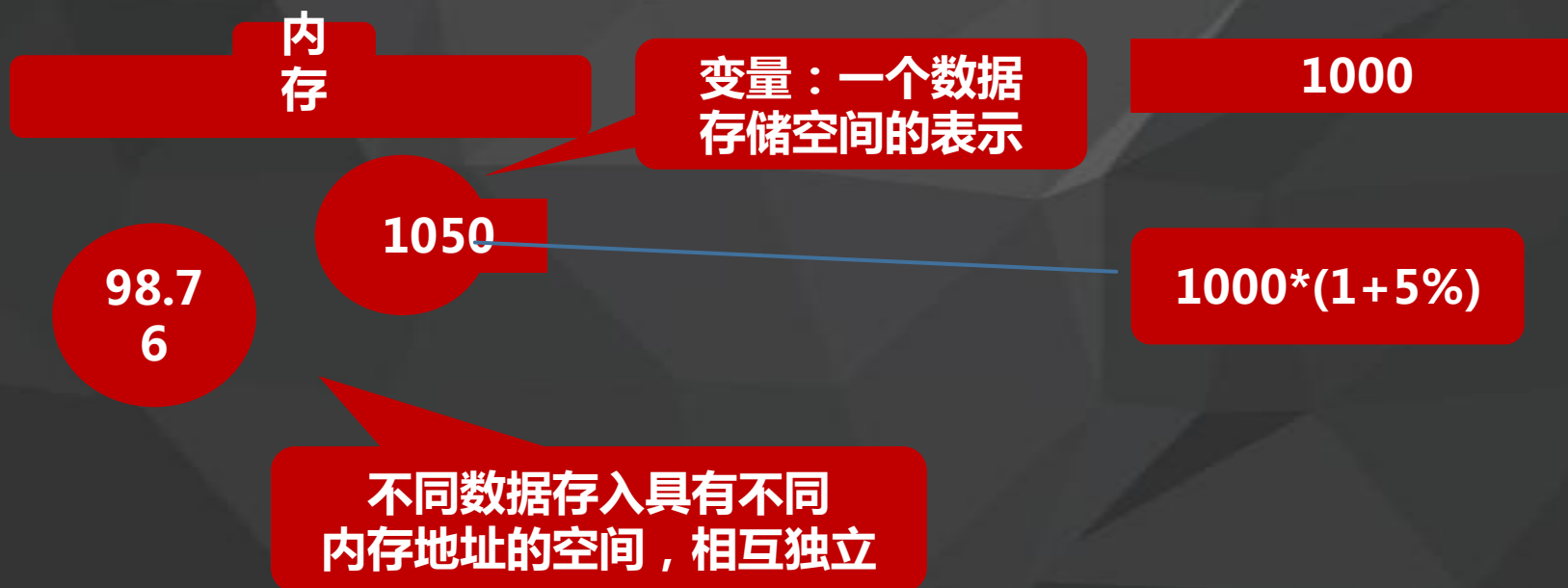
- `int i = 0;`
- `double pi = 3.1415926;`

## ▶ 变量的声明：

- `i = 100;`
- `pi = 3.2;`



- 1、在银行存1000元钱，银行一年的利息5%，那一年之后钱变成了多少？



已经将数据存入内存  
但是：怎么找到存入的数据？

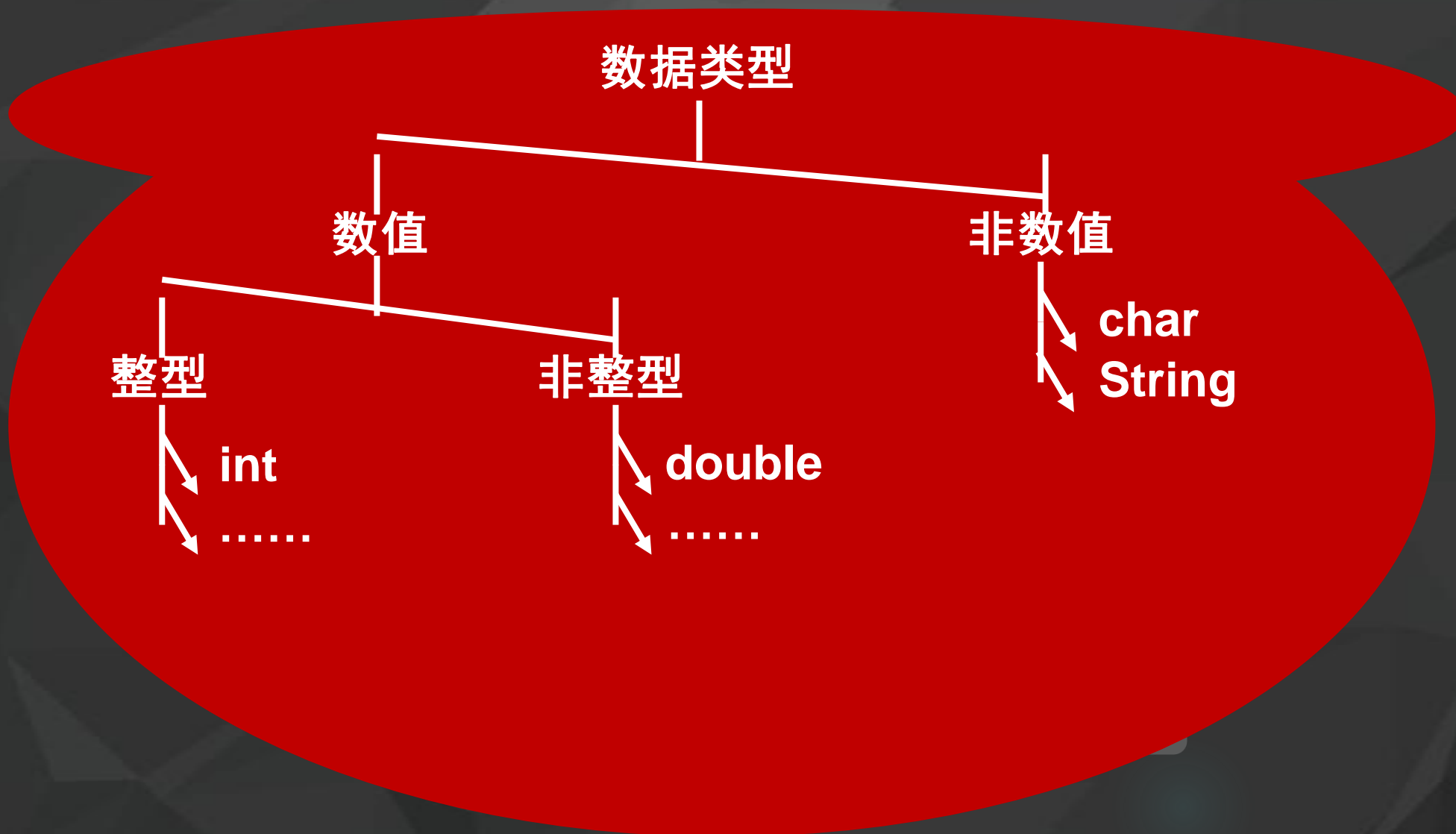
- ▶ 内存地址不好记，怎么办？
  - ▶ 通过内存中小房间的别名找到数据存储的位置



- ▶ 通过变量名可以简单快速地找到它存储的数据

```
int age = 21;
```

# 变量 - 数据类型

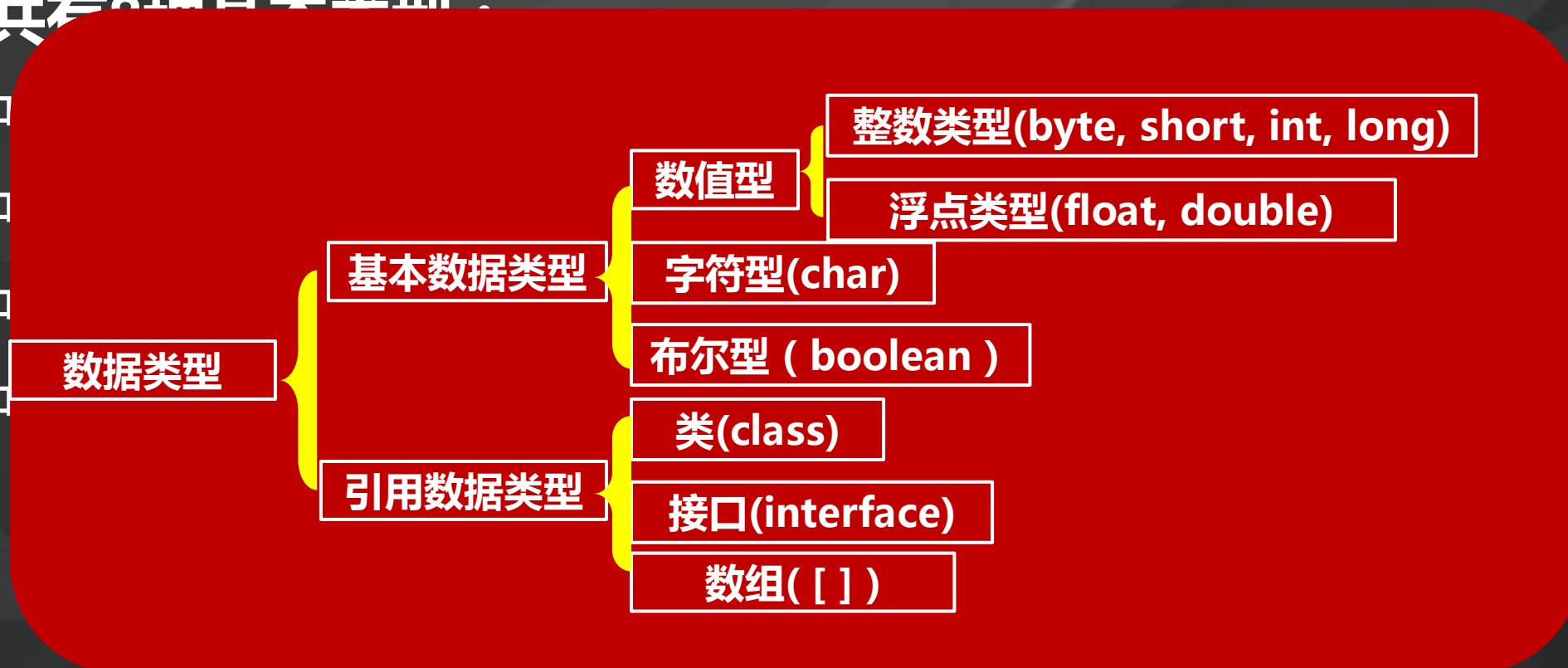




# 数据类型

- ▶ Java是一种强类型语言
- ▶ 每个变量都必须声明类型
- ▶ Java共有8种基本类型。

- ▶ 4种
- ▶ 2种
- ▶ 1种
- ▶ 1种



# 数据类型 - 整形

- ▶ 整型表示无小数部分的数字，允许是负数
- ▶ Java提供了4种整数类型byte、short、int、long
- ▶ 通常情况下，int类型是最常用的

类型	存储条件	范围
byte	1字节	-128~127
short	2字节	-32768~32767
int	4字节	-2147483648~2147483647
long	8字节	-92233720368547758008L~92233720368547758007L

# ◆ 数据类型 - 浮点类型

- ▶ 浮点类型表示有小数部分的数字
- ▶ Java 中有两种浮点类型 float和double
- ▶ 表示float类型数据时需在后面添加后缀F，比如，3.402F
- ▶ 没有后缀F的浮点数据(比如3.402)总被认为是double类型的
- ▶ 当然，也可以在其后面加上后缀D，但不是必须的

类型	存储要求	范围
float	4字节	$\pm 3.40282347E+38F$
double	8字节	大约 $\pm 1.79769313486231570E+308$ (有效数字为6位)

# ◆ 数据类型 - char,boolean

- ▶ char类型占2字节
- ▶ char sex = '男';
- ▶ boolean类型占1字节
- ▶ boolean isWin = false;

注意字符是单引号

逻辑判断值只有两个：  
true和false

boolean isTrue = true;	true
isTrue = 10 > 20;	false
isTrue = 50 == 39;	false

int a = 23;	
int b = 34;	
isTrue = a > b;	false
isTrue = a < b && a == b;	false
isTrue = a < b    a == b;	true

# 常见错误



```
public class Error1 {  
    public static void main(String[] args) {  
        String title;  
        System.out.println(title);  
    }  
}
```

先声明变量并赋值，  
然后才能使用

代码错误

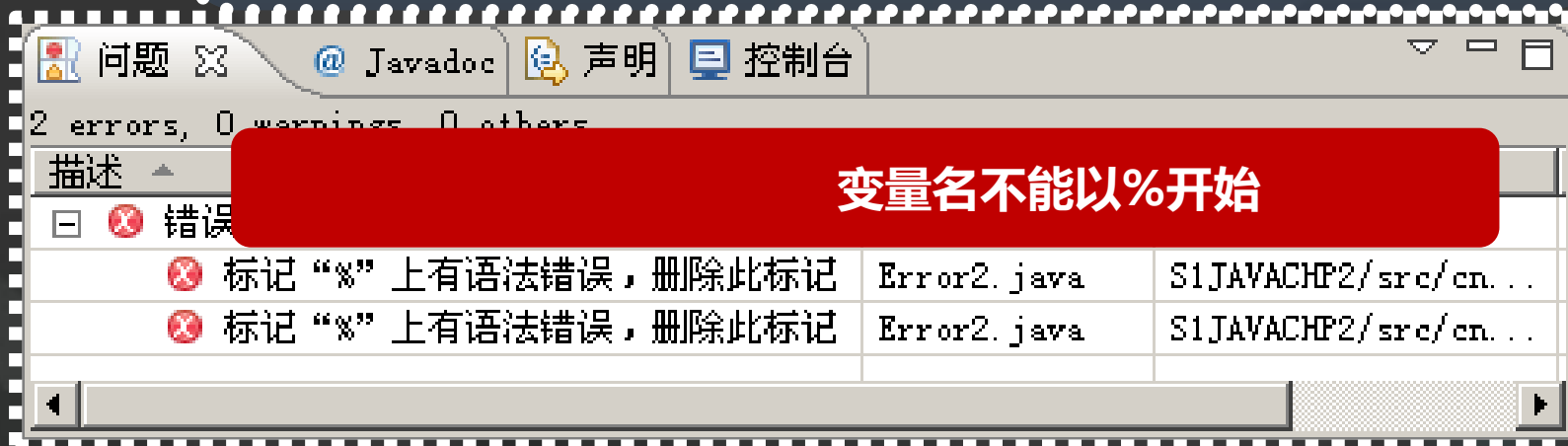


# 常见错误



```
public class Error2 {  
    public static void main(String[] args) {  
        int %hour = 18;  
        System.out.println(%hour);  
    }  
}
```

代码错误



变量名不能以%开始

# 常见错误



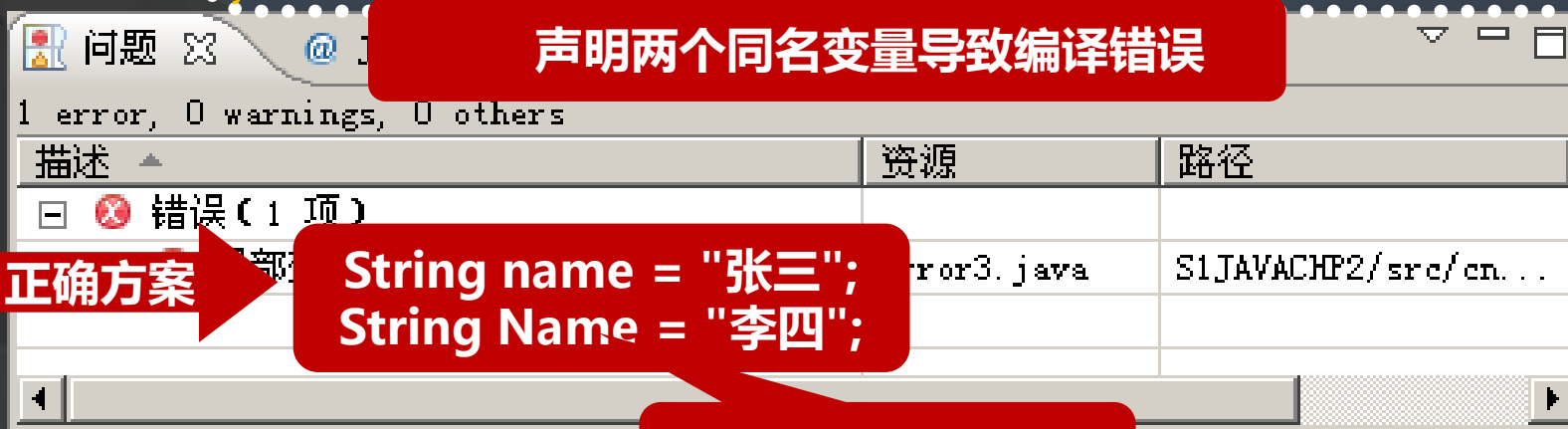
```
public class Error3 {  
    public static void main(String[] args) {  
        String name = "张三";  
        String name = "李四";  
    }  
}
```

声明两个同名变量导致编译错误

正确方案

String name = "张三";  
String Name = "李四";

Java程序区分大小写





- 使用变量存储以下MP3信息，并打印输出
  - 品牌（brand）：爱国者F928
  - 重量（weight）：12.4
  - 电池类型（type）：内置锂电池
  - 价格（price）：499



- ▶ 标识符
- ▶ 关键字
- ▶ 变量
- ▶ 基础数据类型

- ▶ 赋值运算符
- ▶ 算术运算符
- ▶ 关系运算符
- ▶ 逻辑运算符
- ▶ 三元运算符
- ▶ 位运算符

# 运算符 - 赋值运算符



变量名 = 表达式;

## 表达式举例

$a = (b + 3) + (b - 1)$

右边的值给左边的变量

表达式

表达式就是符号（如加号、减号）与操作数（如b，3等）的组合

# 运算符 - 赋值运算符

赋值运算符

运算符	运算	范例	结果
=	赋值	a=3;b=2;	a=3;b=2;
+=	加等于	a=3;b=2;a+=b;	a=5;b=2;
-=	减等于	a=3;b=2;a-=b;	a=1;b=2;
*=	乘等于	a=3;b=2;a*=b;	a=6;b=2;
/=	除等于	a=3;b=2;a/=b;	a=1;b=2;
%=	模等于	a=3;b=2;a%=b;	a=1;b=2;

## ► 注1：赋值运算符注意

► 在JAVA里可以把赋值语句连在一起

►  $x = y = z = 5$

## ► 注2：x += 3等效于x = x + 3，等于\*=，-=，/=依此类推

# 运算符 - 算术运算符

算术运算符

运算符	运算	范例	结果
+	正号	+3	3
-	负号	b=4;-b;	-4
+	加	5+5	10
-	减	6-4	2
*	乘	3*4	12
/	除	5/5	1
%	取模	5%5	0
++	自增 (前)	a=2;b=++a;	a=3;b=3
++	自增 (后)	a=2;b=a++;	a=3;b=2
--	自减 (前)	a=2;b=--a	a=1;b=1
--	自减 (后)	a=2;b=a--	a=1;b=2
+	字符串相加	"He"+"llo"	"Hello"

# 运算符 - 算术运算符



下面代码片断的输出结果是什么？

```
int num1 = 5;
```

% : 求余数

```
int num2 = 2;
```

```
int a = num1 % num2;
```

/ : 求商

```
int b = num1 / num2;
```

输出

5 % 2 = 1

5 / 2 = 2

```
System.out.println(num1 + " % " + num2 + " = " + a);
```

```
System.out.println(num1 + " / " + num2 + " = " + b);
```

```
num1++;
```

等价于 : num1 = num1 + 1;

```
num2--;
```

```
System.out.println("num1 = " + num1);
```

输出

num1 = 6

num2 = 1

```
System.out.println("num2 = " + num2);
```

等价于 : num2 = num2 - 1;

# 案例练习

▶ 结果？

```
int i = 0;  
int j = i+++i;  
System.out.println(j);
```

1

▶ 结果？

```
i = 0;  
j = i++ + ++i;  
System.out.println(j);
```

2

▶ 结果？

```
i = 1;  
j = i+++i++ + ++i;  
System.out.println(j);
```

7



## 1、根据天数（46）计算周数和剩余的天数

提示：

```
int days = 46; // 天数
```

```
int week = days / 7; // 星期
```

```
int leftDay = days % 7; // 剩余的天数
```

## 2、已知圆的半径radius= 1.5，求其面积

提示：

```
double pi = 3.14159; // 圆周率
```

```
double radius = 1.5; // 半径
```

```
double area = pi * radius * radius; // 计算圆面积
```



# ◆ 运算符 - 关系运算符

## ▶ 常用的关系运算符有哪些：

▶  $>$  ,  $<$

▶  $==$  ,  $!=$

▶  $>=$  ,  $<=$

张三的成绩  $>$  李四的成绩  
大象的寿命  $<$  乌龟的寿命  
篮球的大小  $==$  地球的大小

假  
真  
假



由此看出

关系运算符的作用：用来做比较运算  
比较的结果：boolean类型

# 运算符 - 关系运算符

比较运算符

运算符	运算	范例	结果
==	相等于	4==3	false
!=	不等于	4!=3	true
<	小于	4<3	false
>	大于	4>3	true
<=	小于等于	4<=3	false
>=	大于等于	4>=	false
instanceof	检查是否是类的对象	"Hello" instanceof String	true

注1：比较运算符的结果都是boolean型 ( true , false )

注2：比较运算符 "==" 不能误写成 "="

# 运算符 - 逻辑运算符

- 逻辑运算符用于对boolean型结果的表达式进行运算，运算的结果都是boolean型

逻辑运算符

运算符	运算	范例	结果
&	AND (与)	false&true	false
	OR (或)	false true	true
^	XOR (异或)	true^false	true
!	Not (非)	!true	false
&&	AND (短路)	false&&true	false
	OR (短路)	false  true	true

# 运算符 - 逻辑运算符

- ▶ “&” 和 “&&” 的区别：
  - ▶ &&时，如果左边为真，右边参与运算，如果左边为假，右边不参与运算
  - ▶ &时，左边无论真假，右边都进行运算
- ▶ “|” 和 “||” 的区别同理，双或时，左边为真，右边不参与运算
- ▶ 异或( ^ )与或( | )的不同之处是：当左右都为true时，结果为false。

逻辑运算符

运算符	运算	范例	结果
&	AND (与)	false&true	false
	OR (或)	false true	true
^	XOR (异或)	true^false	true
!	Not (非)	!true	false
&&	AND (短路)	false&&true	false
	OR (短路)	false  true	true

# ◆ 运算符 - 三元运算符

## ▶ 格式

- ▶ (条件表达式)?表达式1 : 表达式2 ;
- ▶ 如果条件为true , 运算后的结果是表达式1 ;
- ▶ 如果条件为false , 运算后的结果是表达式2 ;

## ▶ 示例

- ▶ 获取两个数中大数。
- ▶ `int x=3,y=4,z;`
- ▶ `z = (x>y)?x:y;`//z变量存储的就是两个数的大数。

# 运算符 - 位运算符

## ▶ 位运算是直接对二进制进行运算

位运算符		
运算符	运算	范例
<<	左移	$3 < < 2 = 12 \rightarrow 3 * 2 * 2 = 12$
>>	右移	$3 > > 1 = 1 \rightarrow 3 / 2 = 1$
>>>	无符号右移	$3 > > > 1 = 1 \rightarrow 3 / 2 = 1$
&	与运算	$6 \& 3 = 2$
	或运算	$6   3 = 7$
^	异或运算	$6 \wedge 3 = 5$
~	反码	$\sim 6 = -7$

# 运算符 - 位运算符

## 位运算符细节

&lt;&lt;

空位补0，被移除的高位丢弃

&gt;&gt;

被移走的二进制最高位是0，右移后，空缺位补0；  
最高位是1，最高位补是1.

&gt;&gt;&gt;

被移位二进制最高位无论是0或者是1，孔雀尾都用0补。

&amp;

任何二进制位和0进行&运算，结果是0  
和1进行&运算结果是原值

^

任何二进制位和0进行|运算，结果是原值  
和1进行|运算结果是1

~

任何相同二进制位进行^运算，结果是0  
不相同二进制位^运算结果1

# ◆ 小结

- ▶ 标识符
- ▶ 关键字
- ▶ 变量
- ▶ 基础数据类型
- ▶ 运算符

附：练习(变量和运算).doc



# 练习 3

## ▶ 已知两个整形变量如下：

▶ `int a = 55;`

▶ `int b = 22;`

## ▶ 代码实现以下两个变量内数据的交换

代码

阅读

```
int num1 = 19;  
int num2 = 89;  
int temp;  
temp = num1;  
num1 = num2;  
num2 = temp;  
System.out.println(num1);  
System.out.println(num2);
```

借助一个中间变量互换  
两个变量的值

89  
19

交换前 a = 55

交换前 b = 22

交换后 a = 22

交换后 b = 55

# 谢 谢

